目 录

[一、任务描述 - 3 -](#_Toc201078633)

[1.1 歌唱合成（Singing Voice Synthesis, SVS） - 3 -](#_Toc201078634)

[1.2 语音合成（Text-to-Speech, TTS） - 4 -](#_Toc201078638)

[二、数据集介绍 - 4 -](#_Toc201078639)

[三、评价指标介绍 - 5 -](#_Toc201078643)

[3.1 主观评价指标 - 5 -](#_Toc201078644)

[3.2 客观评价指标 - 6 -](#_Toc201078647)

[四、实验结果分析 - 7 -](#_Toc201078650)

[4.1 歌唱合成（SVS）任务 - 7 -](#_Toc201078651)

[4.2 语音合成（TTS）任务 - 13 -](#_Toc201078655)

[五、心得体会 - 17 -](#_Toc201078658)

[5.1 扩散模型版本迭代价值 - 17 -](#_Toc201078659)

[5.2 未来版本优化 - 17 -](#_Toc201078660)

[参考文献 - 17 -](#_Toc201078661)

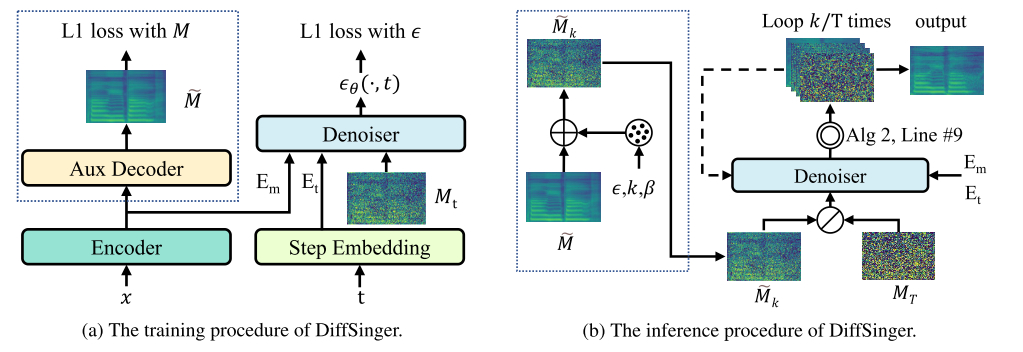
[开源代码的地址 - 17 -](#_Toc201078662)

AI语音处理之歌唱合成与语音合成实验报告

姓名：苗志硕 学号：3220241015

# 一、任务描述

本次实验聚焦于语音合成（TTS）和歌唱合成（SVS）两大任务。语音合成旨在将文本高效转化为自然流畅的语音，而歌唱合成侧重于依据音乐记谱生成具有表现力的歌唱语音。这两个任务在语音生成领域极具挑战性，要求生成的语音不仅要准确传达信息，还要具备自然度、相似度和情感表现力。



#### 图1-1 DiffSinger概述（虚线框中有浅层扩散机制）

在子图（a）中，x为乐谱，t为阶数，M为真谱图；表示经过L1损耗训练的辅助解码器生成的模糊梅尔谱图；Mt是扩散过程第t步的M；在子图（b）中，MT表示第t步扩散处的M（高斯白噪声），K为预测相交边界；有一个选择MT（基础版本）或（浅扩散）作为推理过程的起点的开关。

## 1.1 歌唱合成（Singing Voice Synthesis, SVS）

歌唱合成任务旨在根据乐谱（包括歌词、音高和时长信息）生成自然且富有表现力的歌唱语音。DiffSinger作为基于浅扩散机制的歌唱合成模型，前端处理将歌词转换为语言表示，并结合MIDI信息生成带音高和时长的特征，其核心在于通过扩散概率模型生成声学特征（如梅尔频谱图），再通过声码器转换为波形音频。该代码中提及的DiffSinger存在多种技术实现版本，其核心差异体现在扩散机制优化、模型架构设计及数据集适配策略上。

### 1.1.1基础Naive扩散模型

核心是直接采用标准扩散模型流程，从高斯白噪声开始迭代T=100 步完成梅尔频谱图生成；训练方式是通过优化变分下界（ELBO）最小化噪声预测误差，避免GAN的训练不稳定性；其中推理阶段需从完全噪声开始。

### 1.1.2带浅扩散机制模型

创新点是通过寻找真实频谱M与简单解码器输出的扩散轨迹交点，从浅步骤k（而非T=100）开始反向过程；通过利用辅助解码器（如FastSpeech 2架构）生成，然后通过边界预测网络自适应确定浅步骤k（实验中k=54），推理步骤从k步开始，仅需k次迭代即可完成生成；该方法在保证质量的同时加速推理 45.1%，并提升MOS得分0.14。

### 1.1.3对于不同数据集的版本

（1）PopCS数据集：假设已知真实F0，专注于频谱建模，采用L1损失预训练辅助解码器，结合MFA对齐歌词与音频。

（2）OpenCpop数据集：利用MIDI标签补充旋律前端，包括级联，端到端和PNDM版本。

## 1.2 语音合成（Text-to-Speech, TTS）

语音合成任务是将文本转换为自然语音的过程。DiffSpeech是DiffSinger在TTS任务上的扩展，其核心改进在于使得输入适配，将乐谱输入替换为文本输入，新增音高预测器和时长预测器；模型调整，采用k=70的浅扩散机制，适配TTS任务的特征分布；以及使用基于LJSpeech英文数据集，它也有使用PNDM的版本。

# 二、数据集介绍

## 2.1 OpenCpop数据集（用于SVS）

OpenCpop是一个公开的中文歌唱数据集，包含歌词、MIDI 音符、音符时长等关键信息，有助于提升歌唱合成的质量和表现力，由Wenet团队发布于2022年1月20日。用于训练和测试基于MIDI的歌唱合成模型，支持音高预测和时长预测。

## 2.2 LJSpeech数据集（用于TTS）

LJSpeech是一个英文语音数据集，包含13,100条音频片段（约24小时），含文本转录，由Keith Ito整理发布。使用MFA提取基频（F0）和时长信息，采用FastSpeech 2的预处理流程，包括音素转换和梅尔频谱提取，用于验证DiffSinger在TTS任务的泛化能力。

## 2.3 PopCS数据集（用于SVS）

由于没有公开可用的高质量无伴奏歌唱数据集，它是该论文中收集并注释的一个中文普通话流行歌曲数据集，包含117首中文流行歌曲，约5.89小时，16-bit/24kHz录音，经MFA工具实现音素级对齐，Parselmouth提取 F0 特征，最终整理为1,651个10-13秒的音频片段。该实验随机选择2首歌曲进行验证和测试。

# 三、评价指标介绍

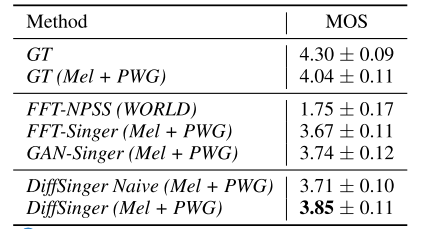
## 3.1 主观评价指标

### 3.1.1 平均意见得分（MOS）

MOS是通过听众对合成音频的自然度、清晰度进行主观评分（1-5分），直观反映语音质量。通过邀请18名合格听众对不同模型生成的歌唱样本进行评分，对比真实音频（GT）、传统模型（FFT-Singer、GAN-Singer）及DiffSinger各版本。

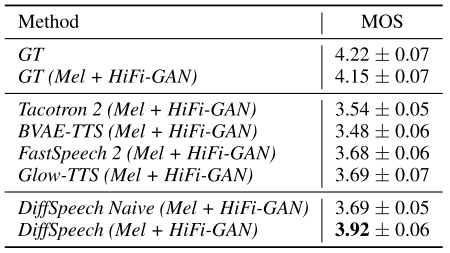
GT（真实音频）为4.30±0.09；DiffSinger带浅扩散机制在歌唱合成任务中的MOS分数为3.85 ± 0.11，相比FFT-Singer（3.67 ± 0.11）和GAN-Singer（3.74 ± 0.12）有显著提升。这表明DiffSinger能生成更自然、高质量的歌声。

在语音合成任务中，DiffSpeech的MOS分数为3.92 ± 0.06，优于FastSpeech 2（3.68 ± 0.06）和Glow-TTS（3.69 ± 0.07），体现了其在语音合成领域的优势。MOS的提升反映了模型在捕捉语音细节、情感表达以及整体自然度方面的能力增强。



#### 图3-1 歌曲样本的最小置信区间为95%

DiffSinger Naive是指没有浅扩散机制的DiffSinger基础版本。



#### 图3-2 具有95%置信区间的语音样本的MOS

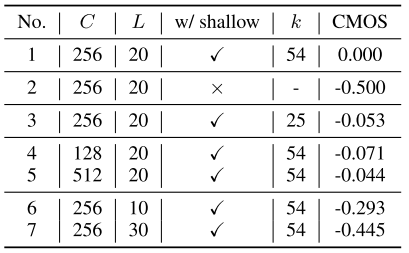
### 3.1.2 比较性主观评价（CMOS）

CMOS是衡量不同模型合成语音的相对质量差异，用于消融实验中对比模型组件的有效性。

通过听众对比两个系统的输出，给出相对评分（如 + 1 到 - 1 分），重点验证浅扩散机制、边界预测等模块的影响。

移除浅扩散机制导致CMOS下降0.500，验证该机制对质量的提升作用；手动设置非最优边界（k=25）时CMOS下降0.053，证明自适应边界预测的必要性。

CMOS的提高表明模型在生成语音的自然度、相似度等方面的综合表现优于其他模型。



#### 图3-3 DiffSinger的变化

其中，C为通道尺寸，L为去噪器的层数，w/shallow表示具有浅扩散机制，K=45是实验中预测的交叉边界。

## 3.2 客观评价指标

3.2.1 实时因子（RTF）

RTF是指生成1秒音频所需的时间，反映模型推理效率（RTF<1表示实时性能）。测量DiffSinger在不同版本下的推理耗时，对比浅扩散机制的加速效果。

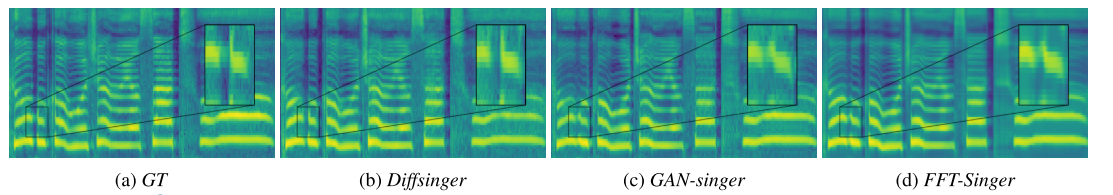
在歌唱合成任务中，DiffSinger的浅层扩散机制使RTF从0.348降至0.191，速度提升了45.1%；在语音合成任务中，DiffSpeech应用浅层扩散机制后，RTF从0.171降至0.121，速度提升了29.2%。通过对比推理耗时，证明浅扩散机制在保持质量的同时显著提升效率。

### 3.2.2 模型参数量（Param）

DiffSinger（SVS）参数量 26.74M，与FFT-Singer（24.25M）接近，优于GAN-Singer（25.217M）。

### 3.2.3 梅尔频谱细节还原能力

通过可视化梅尔频谱图，对比合成音频与真实音频的谐波结构、高频细节。实验绘制绘制GT、DiffSinger、GAN-Singer的梅尔频谱图，观察谐波间的细节差异。DiffSinger生成的频谱图在中低频区域细节更丰富，优于GAN-Singer和FFT-Singer，浅扩散机制通过保留更多初始特征，避免传统模型的过平滑问题。通过可视化对比，直观展示DiffSinger在频谱细节上的优势。



#### 图3-4 四种系统GT，DiffSinger，GAN-Singer和FFT-Singer的MEL谱图的可视化

# 四、实验结果分析

## 4.1 歌唱合成（SVS）任务

在本实验中，基于数据集OpenCpop，由于级联预训练模型太大，这里仅仅给出实现端到端版本的结果。使用预训练模型从输入，其包括文本，注释，持续时间，单词等等，现在取一段音频展示如下，

从打包的测试集进行推理，得到的结果如下：

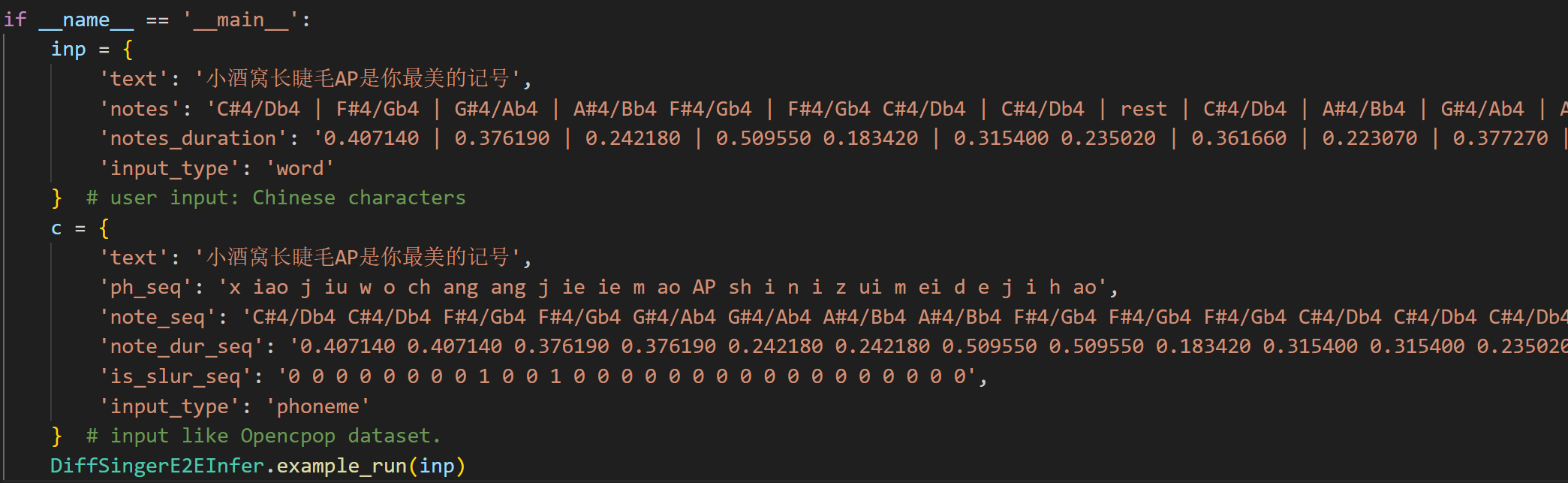
歌声如下：

<https://github.com/Wansui233/zhineng/blob/main/yuyin/%5B2093003457%5D%5BP%5D%E5%B0%8F%E9%85%92%E7%AA%9D%E9%95%BF%E7%9D%AB%E6%AF%9B%E6%98%AF%E4%BD%A0%E6%9C%80%E7%BE%8E%E7%9A%84%E8%AE%B0%E5%8F%B7-0228_opencpop_ds100_rel.wav>

对应的MEL谱图如下：

#### 图4-1 打包测试集推理结果MEL谱图

从原始输入推理：



#### 图4-2 原始输入表示

输出音乐如下：

<https://github.com/Wansui233/zhineng/blob/main/yuyin/example_out.wav>

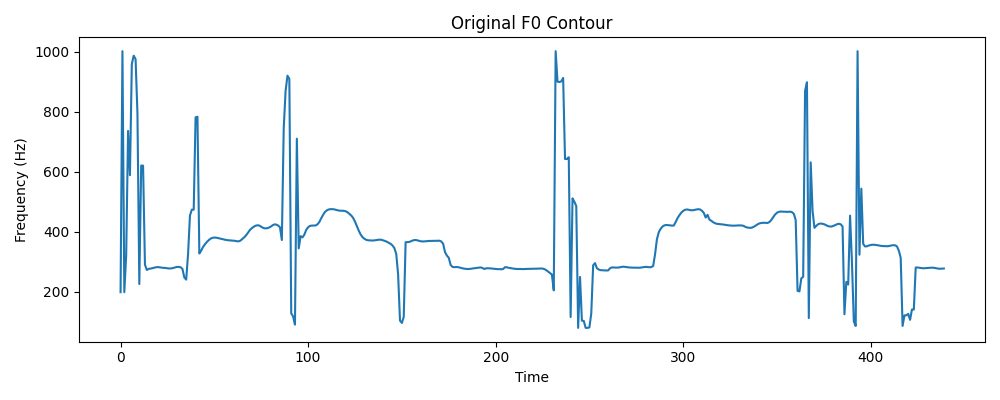
并对该实验得到的音频，为评估实验效果，我们为其写一个评估代码，实现得到输入和输出的MCD，F0 MAE，F0 RMSE，SNR，以及F0图，MEL谱图和波形图。评估代码如下：

<https://github.com/Wansui233/zhineng/blob/main/yuyin/pinggu_SVS_e2e>

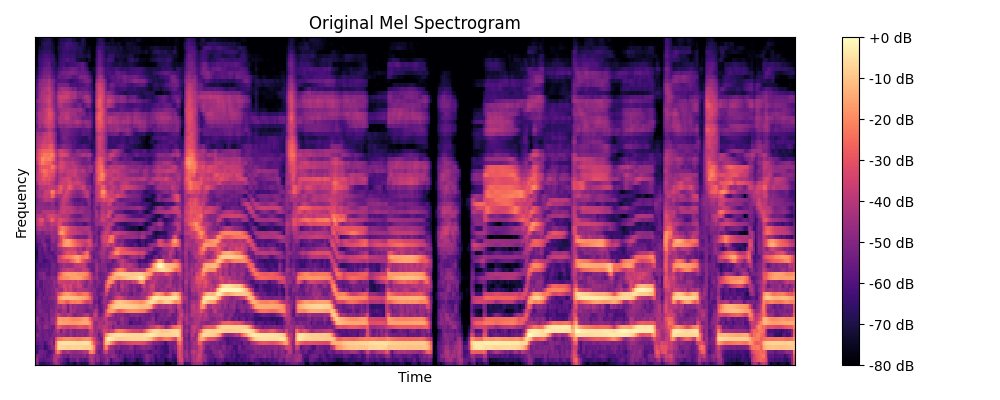
对于从打包的测试集进行推理，得到的结果如下：



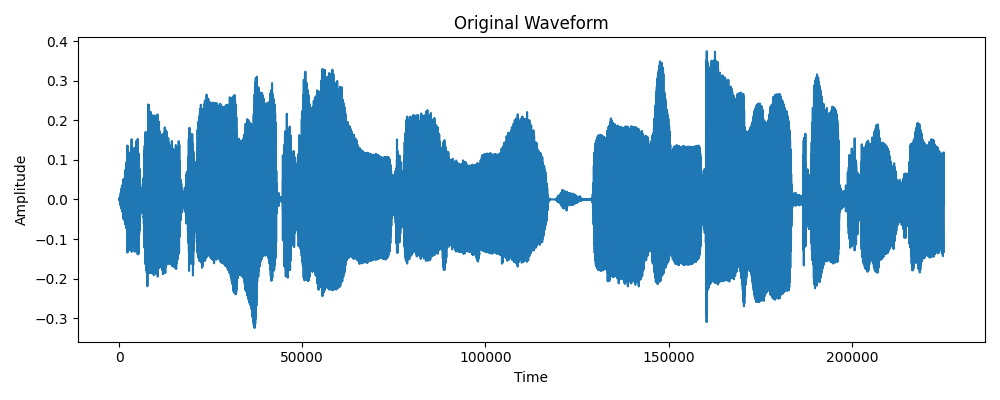
#### 图4-3 从打包测试集推理输出评价对比结果



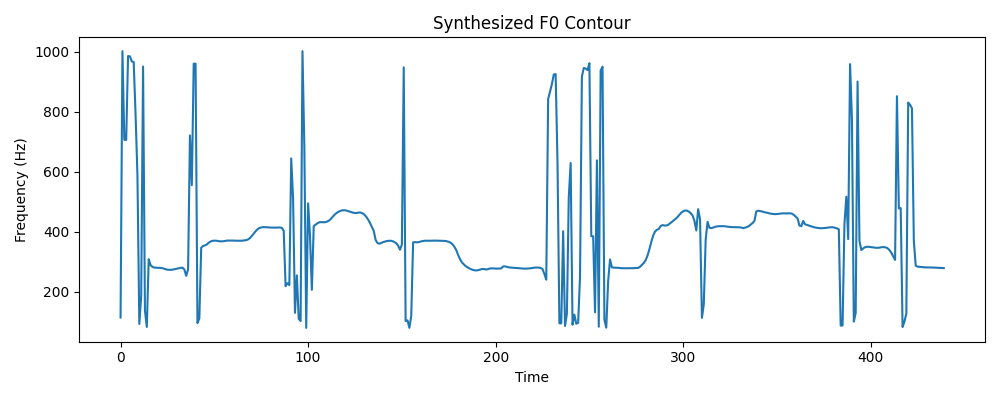
#### 图4-4 数据集中对应音频F0图



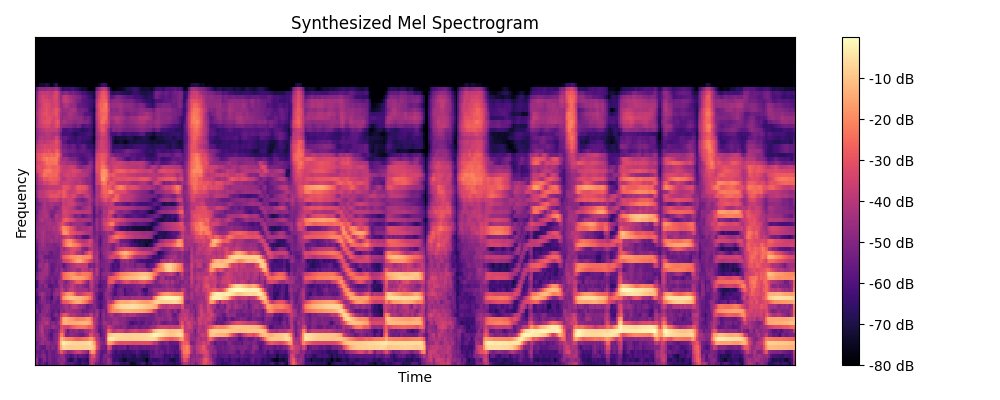
#### 图4-5 数据集中对应MEL谱图



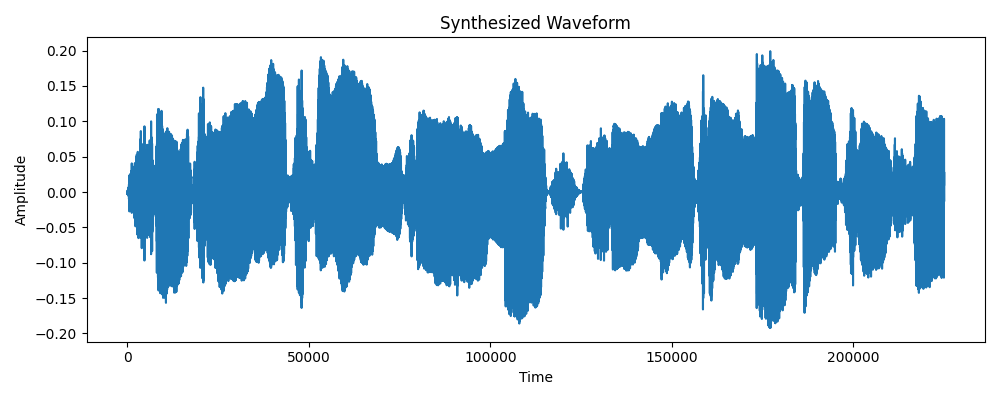
#### 图4-6 数据集中对应波形图



#### 图4-7 打包测试集推理输出对应音频F0图



#### 图4-8 打包测试集推理输出对应MEL谱图

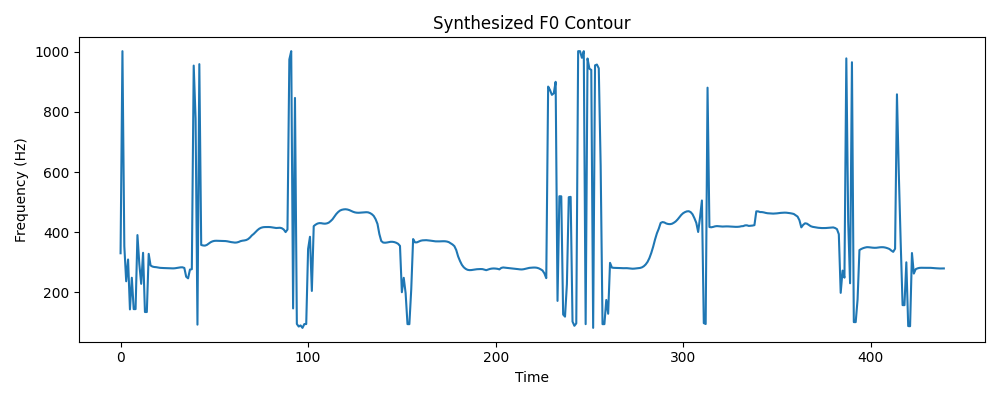


#### 图4-9 打包测试集推理输出对应波形图

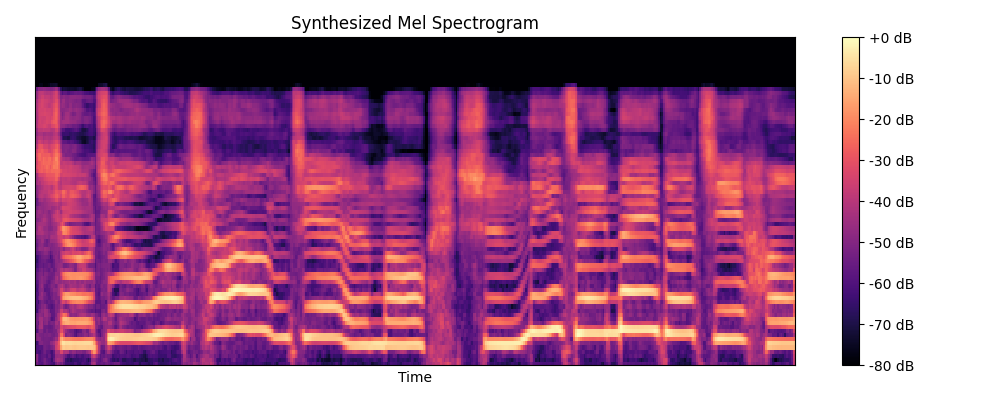
对于从原始输入进行推理，得到的结果如下：



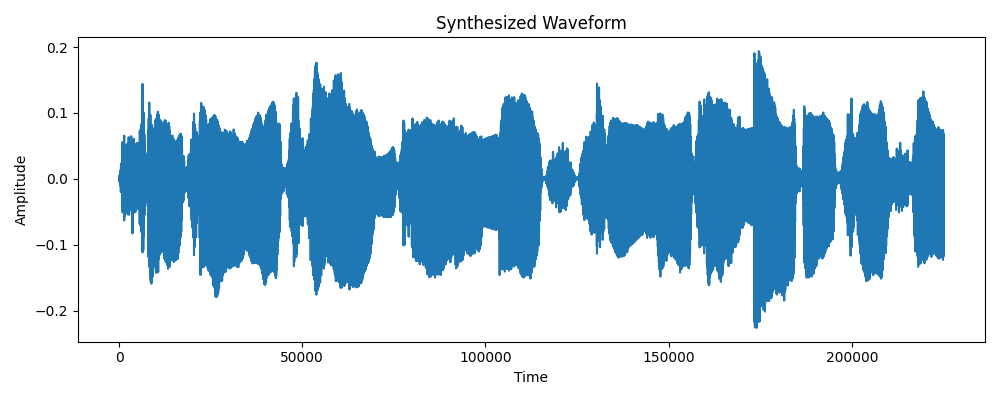
#### 图4-10 从原始输入推理的输出评价对比结果



#### 图4-11 原始输入推理输出对应音频F0图



#### 图4-12 原始输入推理输出对应MEL谱图



#### 图4-13 原始输入推理输出对应波形图

综上所述，该模型在SVS方面获得较好的效果。

### 4.1.1 DiffSinger与基线模型

DiffSinger在MOS评分上显著优于基线系统FFT-Singer和GAN-Singer。DiffSinger的MOS分数为3.85 ± 0.11，相比FFT-Singer 提升了0.18分，比GAN-Singer提升了0.11分。这表明DiffSinger基于扩散概率模型的方法能更好地捕捉歌唱语音的细节和情感，生成更自然、高质量的歌声。从频谱图对比来看DiffSinger生成的频谱图在低频和中频区域细节更丰富，与真实歌声的频谱特征更接近，而GAN-Singer和FFT-Singer在高频部分的表现稍逊色。

### 4.1.2 浅层扩散机制的效果

开启浅层扩散机制后，DiffSinger的MOS分数从3.71提升至3.85，CMOS增加了0.14分，同时速度提升了45.1%（RTF从0.348降至0.191）。这说明浅层扩散机制有效利用了先验知识，为扩散模型提供了更好的起始点，减轻了逆过程的负担，在提升语音质量的同时大幅加快了推理速度。

### 4.1.3 级联与端到端版本

级联版本通过旋律前端显式预测F0曲线和音素时长，在一定程度上实现了基于MIDI信息的歌唱合成。然而，实验发现显式预测F0曲线可能会出现uv/v预测不准确等问题。

端到端版本改进了级联版本的不足，放弃显式预测F0曲线，改由模型联合预测旋律信息，使F0曲线的动态性由生成式模型自动捕捉，简化了流程，提升了生成语音的自然度。

## 4.2 语音合成（TTS）任务

在本实验中，基于数据集LJSpeech，实现DiffSinger的扩展DiffSpeech任务，现取一段音频展示如下：

输入数据：

<https://github.com/Wansui233/zhineng/blob/main/yuyin/LJ001-0001.wav>

测试输出结果：

<https://github.com/Wansui233/zhineng/blob/main/yuyin/%5BLJ001-0001%5D%5BG%5Dprinting%20%2C%20in%20the%20only%20sense%20with%20which%20we%20are%20at%20present%20concerned%20%2C%20differs%20fr-lj_ds_beta6_1213.wav>

#### 图 4-14 对应音频测试输出结果MEL谱图

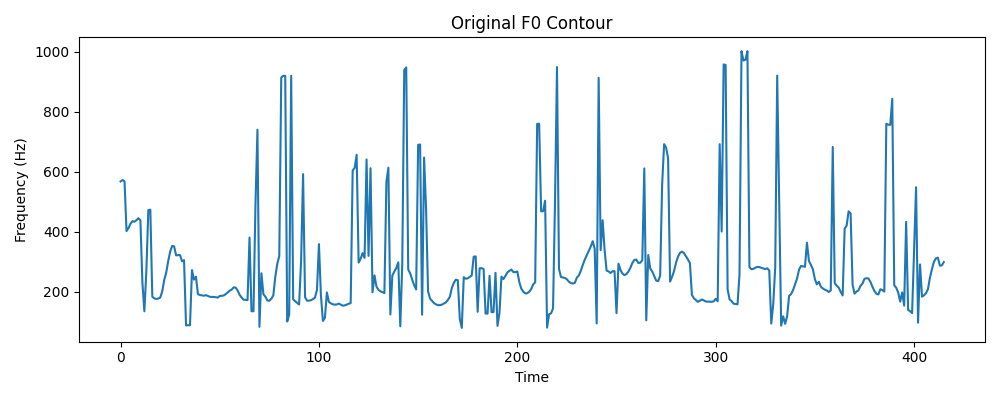
并对该实验得到的音频，为评估实验效果，我们为其写一个评估代码，实现得到输入和输出的MCD，F0 MAE，F0 RMSE，SNR，以及F0图，MEL谱图和波形图。评估代码如下：

<https://github.com/Wansui233/zhineng/blob/main/yuyin/yinpinpinggu_TTS>

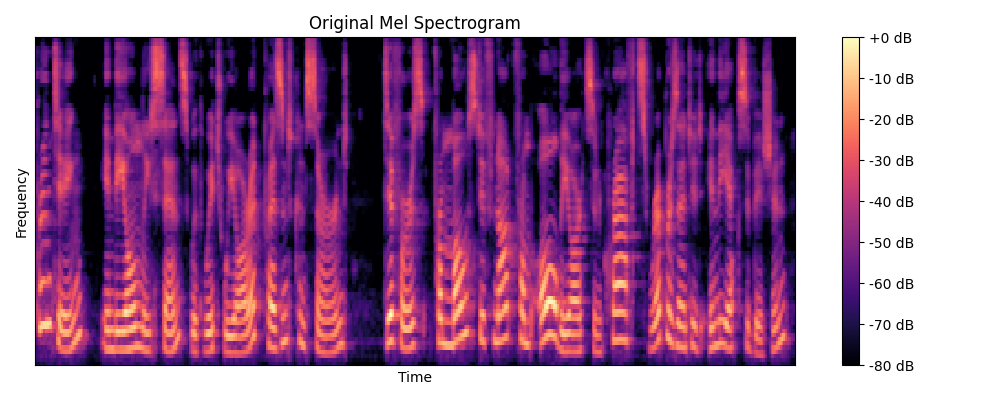
得到的结果如下：

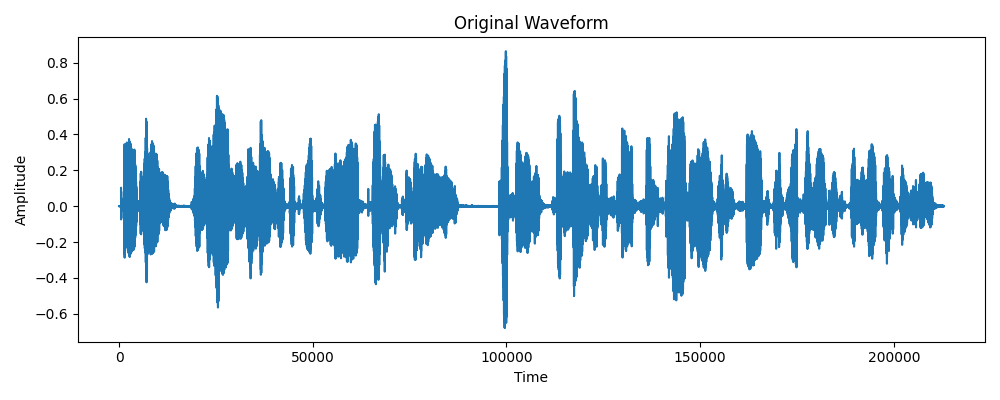
#### 

#### 图4-15 输入与输出评价对比结果

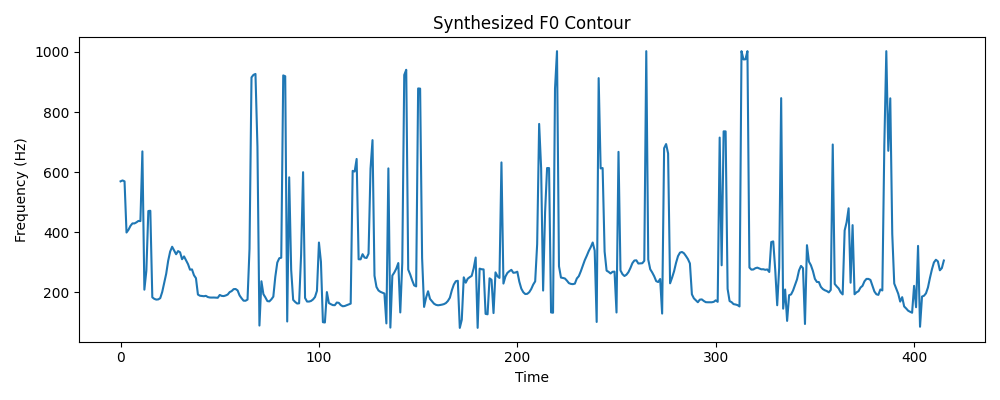


#### 图4-16 数据集对应音频F0图

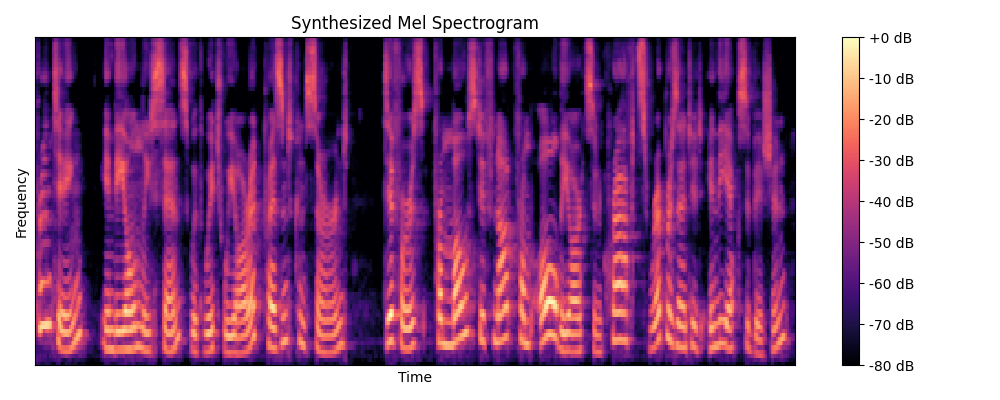
图4-17 数据集对应MEL谱图



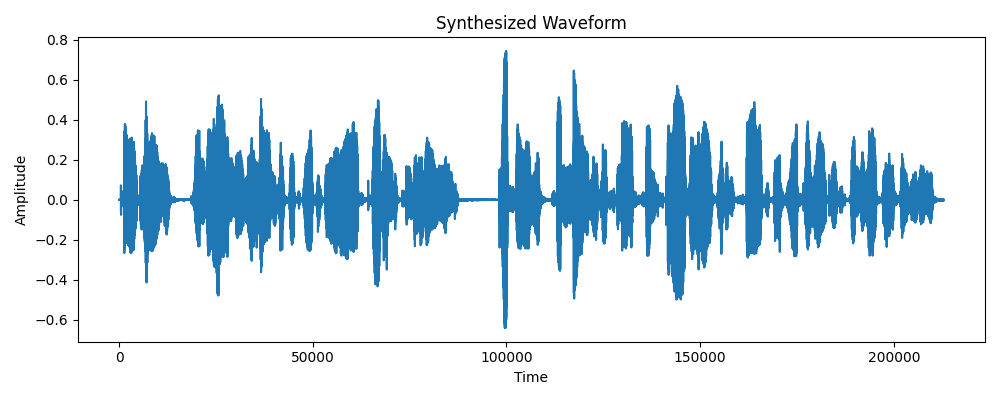
#### 图4-18 数据集对应波形图



#### 图4-19 输出对应音频F0图



#### 图4-20 输出对应音频MEL谱图



#### 图4-21 输出对应音频波形图

综上所述，该模型在TTS方面获得较好的效果。

### 4.2.1 DiffSpeech与基线模型对比

DiffSpeech在MOS评分上优于FastSpeech 2和Glow-TTS。DiffSpeech的MOS分数为3.92 ± 0.06，比FastSpeech 2高0.24分，比Glow-TTS高0.23分。这体现DiffSpeech将扩散模型应用于语音合成任务的优势，能生成更接近真人的语音。频谱可视化显示，DiffSpeech生成的频谱图在整体形态和细节上与真实语音更相似，而FastSpeech 2 的频谱图在高频部分存在一定的模糊和失真。

### 4.2.2 浅层扩散机制的效果

在DiffSpeech中应用浅层扩散机制后，速度提升了29.2%（RTF从0.171降至0.121）。这进一步验证了浅层扩散机制在语音合成任务中也能有效提高推理效率，且在保持语音质量的同时实现加速。

#### 表1 DiffSinger对于SVS任务特点总结

|  |  |  |
| --- | --- | --- |
| 模型版本 | 关键技术特点 | 数据集 |
| DiffSinger Naive | 标准扩散流程，从噪声开始迭代100步 | PopCS |
| DiffSinger浅扩散 | 浅步骤 k=54 开始，边界预测自适应 | PopCS |
| DiffSinger PNDM | 集成渐进非对称扩散，进一步减少迭代步数 | OpenCpop |
| DiffSinger级联 | 分阶段生成（如先F0后频谱） | OpenCpop |
| DiffSinger端到端 | 直接从乐谱到音频，简化流程 | OpenCpop |

#### 表2 扩展TTS任务总结

|  |  |  |  |
| --- | --- | --- | --- |
| 模型版本 | MOS得分 | 与DiffSinger差异 | 数据集 |
| DiffSpeech Naive | 3.69±0.05 | 新增文本前端，k=70 | LJSpeech |
| DiffSpeech浅扩散 | 3.92±0.06 | 文本适配的边界预测 | LJSpeech |
| FastSpeech 2 | 3.68±0.06 | 传统Transformer架构 | LJSpeech |
| Glow-TTS | 3.69±0.07 | 基于流的生成模型 | LJSpeech |

# 五、心得体会

## 5.1 扩散模型版本迭代价值

传统扩散模型从纯噪声生成需数百步，浅扩散通过轨迹交点分析，将起始点提前至k=54步，实现 "热启动" 生成；该机制在不损失质量的前提下，将推理时间缩短近一半，为实时SVS应用（如虚拟歌手）奠定基础。针对OpenCpop数据集设计的级联、端到端版本，反映了不同应用场景的需求。

## 5.2 未来版本优化

开发支持SVS/TTS的统一扩散框架，共享编码器与扩散核心，降低多任务部署成本，论文中DiffSpeech的成功验证了该思路的可行性；通过知识蒸馏将26.74M参数的DiffSinger压缩至10M以内，适配边缘设备等；基于OpenCpop的中文处理经验，开发支持日、韩等东亚语言的SVS版本，重点解决不同语言韵律特征对扩散轨迹交点预测的影响，等等。

综上所述，本实验通过分析DiffSinger在SVS和TTS任务的多个技术版本，验证了扩散模型在语音生成领域的可扩展性，从Naive扩散（基础功能）到浅扩散（效率优化），再到PNDM或级联。DiffSinger和DiffSpeech的创新性浅层扩散机制有效地解决了传统扩散模型速度慢的问题，同时提升了语音质量，在歌唱合成和语音合成任务中均取得了显著优势。

# 参考文献

DOI：<https://arxiv.org/abs/2105.02446>

# 开源代码的地址

<https://github.com/MoonInTheRiver/DiffSinger>